SYDE 372
Introduction to Pattern Recognition

Discriminant Functions: Part I

Alexander Wong

Department of Systems Design Engineering
University of Waterloo

Outline

Motivation

- So far, the approach to the labeled sample problem is:
  - Use the given samples to obtain a class description consisting of either a distance metric or probability density function
  - Derive decision rule from description (e.g., MICD and MAP rules)
- The decision rule in turn specifies a decision boundary in feature space.
- For example, MICD rule and MAP rule have decision surfaces of the form:

$$g(\underline{x}) = \underline{x}^T Q_0 \underline{x} Q_1 \underline{x} + Q_2 = 0 \tag{1}$$

Motivation

- The function $g(\underline{x})$ is a **discriminant function**
- The two-class decision rule can be written as:

$$g(\underline{x}) \underset{B}{\overset{A}{\gtrless}} 0 \qquad (2)$$

- A positive value for $g(\underline{x})$ means that the pattern $\underline{x}$ belongs to class A, while a negative value for $g(\underline{x})$ means that the pattern $\underline{x}$ belongs to class B

Motivation

- Idea: What if we take an alternative approach?
  - Assume a particular form for the discriminant functions (e.g., hyperplane)
  - Use the given samples to directly estimate the parameters of the discriminant functions
  - Given discriminant functions, decision rules and decision surfaces are defined
- What we basically want to do is learn the discriminant functions directly from the samples.

Linear Discriminants

- A linear discriminant function can be expressed as:

$$g(\underline{x}) = \underline{w}^T \underline{x} + w_0 \tag{3}$$

where $\underline{w}$ is the weight vector and w_0 is a threshold.

- If we set $g(\underline{x})$, we have the equation for a hyperplane, with the decision surface defined for a linear classifier.

Linear Discriminants

- A more explicit way of expressing $g(\underline{x})$ to emphasize its linear nature is:

$$g(\underline{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + w_0 \tag{4}$$

$$g(\underline{x}) = \sum_{i=1}^{n} w_i x_i + w_0 \tag{5}$$

- e.g., For a two-class case $\underline{x} = (x_1, x_2)$, the linear discriminant $g(x_1, x_2)$ can be written as:

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2} \tag{6}$$

- Just a straight line equation!

Linear Discriminants

- Consider the two class problem with discriminant $g(\underline{x}) = \underline{w}^T\underline{x} + w_0$.
- The decision rule can be defined as:
  - $\underline{x} \in c_1$ if $g(\underline{x}) > 0$
  - $\underline{x} \in c_2$ if $g(\underline{x}) < 0$
- The decision surface, defined by $g(\underline{x}) = 0$, is a hyperplane with the following properties:
  - The unit normal vector is $\frac{\underline{w}}{|\underline{w}|}$, since for any two vectors $\underline{x}_1$ and $\underline{x}_2$:

$$g(\underline{x}_1) = g(\underline{x}_2) = 0 \tag{7}$$

$$\underline{w}^T\underline{x}_1 + w_0 = \underline{w}^T\underline{x}_2 + w_0 \tag{8}$$

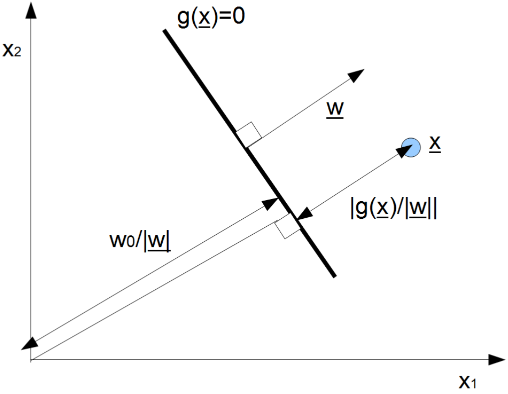
$$\underline{w}^T(\underline{x}_1 - \underline{x}_2) = 0 \tag{9}$$

This shows that $\underline{w}$ is normal to any vector lying in the plane so that $\frac{\underline{w}}{|\underline{w}|}$ is the unit normal.

Linear Discriminants

- The decision surface, defined by $g(\underline{x}) = 0$, is a hyperplane with the following properties:
  - The distance between any $\underline{x}$ and the hyperplane is $\left|\frac{g(\underline{x})}{|\underline{w}|}\right|$
  - When $g(\underline{x}) > 0$, $\underline{x}$ is said to lie on the positive side of the plane, the side which $\underline{w}$ points to.
  - When $g(\underline{x}) < 0$, $\underline{x}$ is said to lie on the negative side of the plane.

Linear Discriminants: Visualization



Linear Discriminants: Example

- Suppose that we are given two classes that are linearly separable with the following discriminant function:

$$g(\underline{x}) = 4\underline{x}_1 + 3\underline{x}_2 - 5 \tag{10}$$

  and the following decision rule
  - $\underline{x} \in c_1$ if $g(\underline{x}) > 0$
  - $\underline{x} \in c_2$ if $g(\underline{x}) < 0$
- For the unit normal vector of the decision boundary and its distance from the origin. Plot the boundary indicating $\underline{w}$ and w_0.
- Classify the following patterns: $\underline{x}_1 = (1, 3), \underline{x}_2 = (2, -1), \underline{x}_3 = (1, -3)$

Linear Discriminants: Example

- If we were to rewrite $g(x)$ in vector form, we end up with:

$$g(\underline{x}) = [4 \;\; 3] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 5 \underset{B}{\overset{A}{\gtrless}} 0 \tag{11}$$

$$g(\underline{x}) = \underline{w}^T \underline{x} + w_0 \tag{12}$$

- Therefore, given this vector form, we know that:

$$\underline{w} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \tag{13}$$

Linear Discriminants: Example

- Therefore, the unit normal vector can be computed as:

$$\frac{\underline{w}}{|\underline{w}|} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} / (\sqrt{4^2 + 3^2}) \tag{14}$$

$$\frac{\underline{w}}{|\underline{w}|} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} / (5) \tag{15}$$

$$\frac{\underline{w}}{|\underline{w}|} = \begin{bmatrix} 4/5 \\ 3/5 \end{bmatrix} \tag{16}$$

Linear Discriminants: Example

- The distance from the origin to the plane is given by:

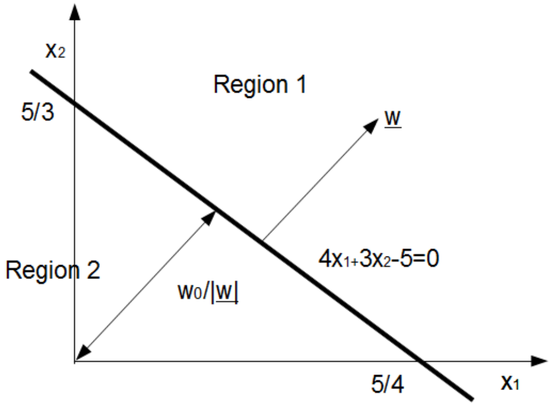
$$d = \left|\frac{g(\underline{x})}{|\underline{w}|}\right| = \left|\frac{g(0,0)}{(\sqrt{4^2+3^2})}\right| \tag{17}$$

$$d = \left|\frac{4(0)+3(0)-5}{(\sqrt{4^2+3^2})}\right| \tag{18}$$

$$d = \left|\frac{-5}{5}\right| \tag{19}$$

$$d = 1 \tag{20}$$

Linear Discriminants: Example



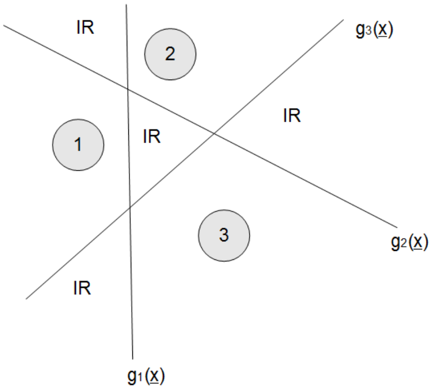
Linear Discriminants: Example

- To classify patterns, we plug our patterns into the decision rule:
  - For $x_1 = (1, 3)$, $g(1, 3) = 4(1) + 3(3) - 5 = 13 - 5 = 8 > 0$, so class=$c_1$
  - For $x_1 = (2, -1)$, $g(2, -1) = 4(2) + 3(-1) - 5 = 5 - 5 = 0 = 0$, so class=$c_1$ or c_2
  - For $x_1 = (91, -3)$, $g(91, -3) = 4(91) + 3(-3) - 5 = 355 - 5 = 350 > 0$, so class=$c_1$

Multi-class classification using linear discriminants

- So far, we've only talked about defining decision regions for the two class problem
- How do we handle the situation where we have multiple classes ($k > 2$)?
- Solution: Use multiple linear discriminants to separate the different classes!
- Three possible strategies:
  - Strategy 1: A linear discriminant can be found for each class which separates it from all other classes:
    - $g_i(\underline{x}) > 0$ if $\underline{x} \in c_i$, $i = 1, \ldots, k$
    - $g_i(\underline{x}) < 0$ if otherwise

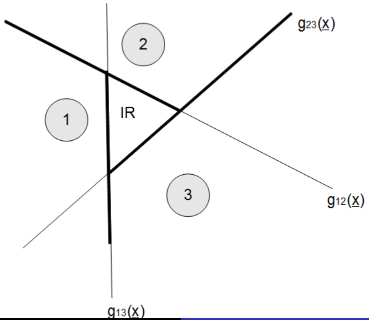
Multi-class classification using linear discriminants: Strategy 1



Quite a few indeterminant areas.

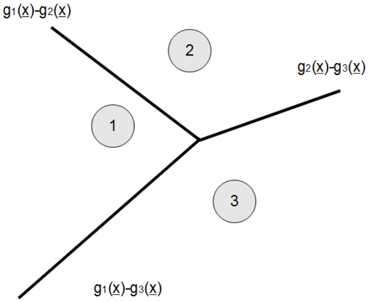
Multi-class classification using linear discriminants

- Three possible strategies:
  - Strategy 2: A linear discriminant can be found for every pair of classes (i.e., classes are pairwise separable):
    - $g_{ij}(\underline{x}) > 0$ for all $j \neq i$ if $\underline{x} \in c_i$



Multi-class classification using linear discriminants

- Three possible strategies:
  - Strategy 3: Each class has its own discriminant function:
    - $g_i(\underline{x}) > g_j(\underline{x})$ for all $j \neq i$ if $\underline{x} \in c_i$



Multi-class classification using linear discriminants

- Of the three strategies, only the last strategy avoids producing indeterminant regions.
- Therefore, adopting this strategy, the general k class linear discriminant classifier can be defined as:

$$\underline{x} \in c_i \text{ iff } g_i(\underline{x}) > g_j(\underline{x}) \text{ for all } j \neq i \tag{21}$$

with $g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{i0}$, $i = 1, \ldots, k$

- Based on this classifier, the decision boundary between c_i and c_j is given by $g_i(\underline{x}) = g_j(\underline{x})$:

$$g_i(\underline{x}) - g_j(\underline{x}) = (\underline{w}_i - \underline{w}_j)^T \underline{x} + w_{i0} - w_{j0} = 0 \tag{22}$$

- This is a hyperplane with normal vector $(w_i - w_j)$!

Learning discriminants

- The question now is: how do we build such a k class linear discriminant classifier?
- Suppose that we are given a set of labeled samples for each of the classes which are assumed to be **linearly separable** in an appropriate feature space.
- The goal is to learning appropriate discriminant functions $g_i(\underline{x})$ directly from the labeled samples
- Focusing on the two-class problem, the problem of learning the discriminant is to find a weight vector $\underline{a}$ such that:
  - $g(\underline{x}) = \underline{a}^T \underline{y} > 0$ when $\underline{y}$ (and $\underline{x}$) is a member of class c_1
  - $g(\underline{x}) = \underline{a}^T \underline{y} < 0$ when $\underline{y}$ (and $\underline{x}$) is a member of class c_2

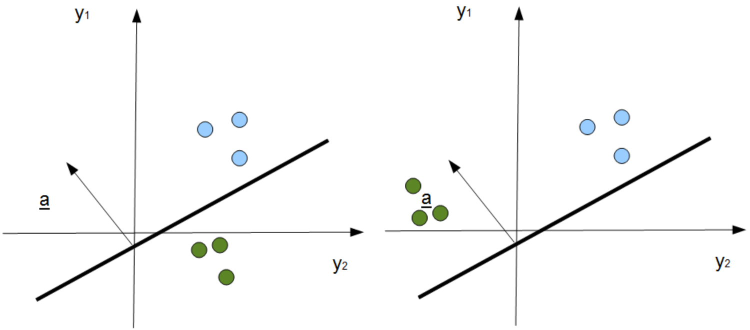
Learning discriminants

- If the classes are linearly separable in the original feature space ($\underline{x}$), we have: $\underline{y} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ 1 \end{bmatrix}$ and $\underline{a} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ w_0 \end{bmatrix}$
- In $\underline{y}$ space, the decision surface is a hyperplane which contains the original and has normal vector $\frac{\underline{a}}{|\underline{a}|}$.
- Given labeled samples $\{y_1, y_2, \ldots, y_N\}$, the goal is to find $\underline{a}$ (the solution vector) such that:
  - $\underline{a}^T \underline{y}_i > 0$ for all $\underline{y}_i \in c_1$
  - $\underline{a}^T \underline{y}_i < 0$ for all $\underline{y}_i \in c_2$

Learning discriminants

- One way to simplify the problem a bit is to perform normalization by replacing all $\underline{y}_i$ by $-\underline{y}_i$ for all $\underline{y}_i \in c_2$.
- By doing so, we can change our goal to finding $\underline{a}^T\underline{y}_i > 0$ for all i! The solution remains the same!

Learning discriminants



Before normalization

After normalization

Learning discriminants

- So how do we find a solution vector $\underline{a}$ that satisfies our classification criteria?
- Trial and error and exhaustive search strategies are impractical for the general N sample n dimensional problem.
- A much more efficient strategy is to use iterative methods that use a criterion function which is minimized when $\underline{a}$ is the solution vector.

Learning discriminants

- Here, we will use gradient descent optimization strategies to find $\underline{a}$:
- Let $J(\underline{a})$ be the criterion function
- The weight vector at $k+1$ ($\underline{a}_{k+1}$) is computed based on the weight vector at k ($\underline{a}_k$) and the gradient of the criterion function ($\nabla J(\underline{a})$).
- Since $\nabla J(\underline{a})$ indicates the direction of maximum change, we wish to move in the opposite direction, which is the direction of steepest descent:

$$\underline{a}_{k+1} = \underline{a}_k - \rho_k \nabla J(\underline{a}) \tag{23}$$

where ρ_k is the step size (which dictates the rate of convergence)

Gradient descent approaches for Learning discriminants

- Here, we will discuss two types of gradient descent approaches for learning discriminants:
  - **Perceptron approach**: guide convergence based on sum of distances of misclassified samples to decision boundary
  - **Minimum Squared Error approach**: guide convergence based on sum of squared error
- Each comes in different varieties!
  - Non-sequential: update based on all samples at the same time
  - Sequential: update based on one sample at a time

Perceptron approach

- The perceptron criterion may be interpreted as the sum of distances of the misclassified samples from the decision boundary.

$$J_p(\underline{a}) = \sum_{y \in Y(\underline{a})} (-\underline{a}^T \underline{y}) \tag{24}$$

where Y is the set of misclassified samples due to $\underline{a}$:

$$Y(\underline{a}) = (y_i \text{ such that } \underline{a}^T \underline{y}_i \leq 0) \tag{25}$$

- $\underline{a}$ is the solution vector when $J_p(\underline{a}) = 0$.

Perceptron approach

- The gradient of $J_p(\underline{a})$ can be written as:

$$\nabla J_p(\underline{a}) = \sum_{y \in Y(\underline{a})} (-\underline{y}) \tag{26}$$

- This gives us the weight update formula as:

$$\underline{a}_{k+1} = \underline{a}_k + \rho_k \nabla J_p(\underline{a}) \tag{27}$$

$$\underline{a}_{k+1} = \underline{a}_k - \rho_k \sum_{y \in Y(\underline{a})} (-\underline{y}) \tag{28}$$

$$\underline{a}_{k+1} = \underline{a}_k + \rho_k \sum_{y \in Y(\underline{a})} (\underline{y}) \tag{29}$$

Perceptron approach

- **Step 1:** Set an initial guess for the weight vector ($\underline{a}_0$) and let $k = 0$
- **Step 2:** Based on $\underline{a}_k$, construct the classifier and determine the set of misclassified samples $Y(\underline{a})$. If there are no misclassified samples, stop here since we have arrived at the solution. Otherwise, continue to Step 3.
- **Step 3:** Compute a scalar multiple of the sum of misclassified samples $\rho_k \sum_{y \in Y(\underline{a})} (\underline{y})$
- **Step 4:** Determine $\underline{a}_{k+1}$ as

$$\underline{a}_{k+1} = \underline{a}_k + \rho_k \sum_{y \in Y(\underline{a})} (\underline{y}) \tag{30}$$

- **Step 5:** Go to Step 2.

Variations on the Perceptron approach

- Fixed-increment: $\rho_k = 1$, constant step size.
- Variable-increment: $\rho_k \propto 1/k$, decreases as number of iterations increases to avoid over-shooting solution.
- Single sample correction: Treat samples sequentially, change weight vector with each misclassification.

Sequential Perceptron approach

- **Step 1:** Set an initial guess for the weight vector ($\underline{a}_0$) and let $k = 0$
- **Step 2:** Based on $\underline{a}_k$, construct the classifier and determine the set of misclassified samples $Y(\underline{a})$. If there are no misclassified samples, stop here since we have arrived at the solution. Otherwise, continue to Step 3.
- **Step 3:** Compute a scalar multiple of the k^{th} misclassified sample $\rho_k \underline{y}^k$
- **Step 4:** Determine $\underline{a}_{k+1}$ as

$$\underline{a}_{k+1} = \underline{a}_k + \rho_k \underline{y}^k \tag{31}$$

- **Step 5:** Go to Step 2.

Perceptron approach: Example

- Suppose that we are given the following data: $y_1 = (4, -1)$, $y_2 = (2, 1)$ belong to class c_1, and $y_3 = (5/2, -5/2)$ belong to class c_2.
- Let the initial guess be $\underline{a}_0 = (0, 0)$ and $\rho_k = 1$ for all k.
- Use the standard perceptron approach to learn a solution vector $\underline{a}$.
- Use the sequential perceptron approach to learn a solution vector $\underline{a}$.

Perceptron approach: Example

- **Step 1:** To simplify the problem, normalize by replacing $\underline{y}_i$ by $-\underline{y}_i$ for all $\underline{y}_i \in c_2$
  - Therefore, in this case, $y_3 = (-5/2, 5/2)$.
- **Step 2:** Based on $\underline{a}_0$, construct the classifier and determine the set of misclassified samples $Y(\underline{a})$.
  - $\underline{a}_0^T \underline{y}_1 = \underline{a}_0^T \underline{y}_2 = \underline{a}_0^T \underline{y}_3 = 0$
  - Since we arrive at our solution when $\underline{a}^T \underline{y}_i > 0$ for all i, all three cases fail!
  - Therefore, set of misclassified samples are $\underline{y}_0, \underline{y}_2, \underline{y}_3$

Perceptron approach: Example

- **Step 3:** Determine $\underline{a}_1$ as

$$\underline{a}_1 = \underline{a}_0 + \sum_{i=1}^{3} (\underline{y}_i) = (7/2, 5/2) \quad (32)$$

  - Let's go back to step 2!
- **Step 2:**
  - $\underline{a}_1^T \underline{y}_1 = [7/2 \;\; 5/2][4 \;\; -1]^T = 23/2 > 0$ (correct)
  - $\underline{a}_1^T \underline{y}_2 = [7/2 \;\; 5/2][2 \;\; 1]^T = 19/2 > 0$ (correct)
  - $\underline{a}_1^T \underline{y}_3 = [7/2 \;\; 5/2][-5/2 \;\; 5/2]^T = -5/2 < 0$ (misclassified)
  - Therefore, set of misclassified samples is $\underline{y}_3$

Perceptron approach: Example

- **Step 3:** Determine $\underline{a}_2$ as

$$\underline{a}_2 = \underline{a}_1 + (\underline{y}_3) \tag{33}$$

$$\underline{a}_2 = [7/2 \;\; 5/2] + [-5/2 \;\; 5/2] = [1 \;\; 5] \tag{34}$$

  - Let's go back to step 2!
- **Step 2:**
  - $\underline{a}_2^T \underline{y}_1 = [1 \;\; 5][4 \;\; -1]^T = -1 < 0$ (misclassified)
  - $\underline{a}_2^T \underline{y}_2 = [1 \;\; 5][2 \;\; 1]^T = 7 > 0$ (correct)
  - $\underline{a}_2^T \underline{y}_3 = [1 \;\; 5][-5/2 \;\; 5/2]^T = 10 > 0$ (correct)
  - Therefore, set of misclassified samples is $\underline{y}_1$

Perceptron approach: Example

- **Step 3:** Determine $\underline{a}_3$ as

$$\underline{a}_3 = \underline{a}_2 + (\underline{y}_1) \tag{35}$$

$$\underline{a}_3 = [1 \;\; 5] + [4 \;\; -1] = [5 \;\; 4] \tag{36}$$

  - Let's go back to step 2!
- **Step 2:**
  - $\underline{a}_3^T \underline{y}_1 = [5 \;\; 4][4 \;\; -1]^T = 16 > 0$ (correct)
  - $\underline{a}_3^T \underline{y}_2 = [5 \;\; 4][2 \;\; 1]^T = 14 > 0$ (correct)
  - $\underline{a}_3^T \underline{y}_3 = [5 \;\; 4][-5/2 \;\; 5/2]^T = -5/2 < 0$ (misclassified)
  - Therefore, set of misclassified samples is $\underline{y}_3$

Perceptron approach: Example

- **Step 3:** Determine $\underline{a}_4$ as

$$\underline{a}_4 = \underline{a}_3 + (\underline{y}_3) \tag{37}$$

$$\underline{a}_4 = [5 \;\; 4] + [-5/2 \;\; 5/2] = [5/2 \;\; 13/2] \tag{38}$$

  - Let's go back to step 2!
- **Step 2:**
  - $\underline{a}_4^T \underline{y}_1 = [5/2 \;\; 13/2][4 \;\; -1]^T = 7/2 > 0$ (correct)
  - $\underline{a}_4^T \underline{y}_2 = [5/2 \;\; 13/2][2 \;\; 1]^T = 23/2 > 0$ (correct)
  - $\underline{a}_4^T \underline{y}_3 = [5/2 \;\; 13/2][-5/2 \;\; 5/2]^T = 10 > 0$ (correct)
  - Therefore, there are no misclassified samples and we stop!

Perceptron approach: Example

- Let's use the sequential perceptron approach!
- Determine $\underline{a}_1$ as

$$\underline{a}_1 = \underline{a}_0 + \underline{y}_1 = (4, -1) \tag{39}$$

- Check if $\underline{y}_2$ is correctly classified
  - $\underline{a}_1^T \underline{y}_2 = [4 \quad -1][2 \quad 1]^T = 7 > 0$ (correct)
- Check if $\underline{y}_3$ is correctly classified
  - $\underline{a}_1^T \underline{y}_3 = [4 \quad -1][-5/2 \quad 5/2]^T = -25/2 < 0$ (misclassified)
- Since $\underline{y}_3$ is misclassified, we compute a_2 as

$$\underline{a}_2 = \underline{a}_1 + \underline{y}_3 = (3/2, 3/2) \tag{40}$$

Perceptron approach: Example

- Check if $\underline{y}_1$ is correctly classified
  - $\underline{a}_2^T \underline{y}_1 = [3/2\ \ 3/2][4\ \ -1]^T = 15/2 > 0$ (correct)
- Check if $\underline{y}_2$ is correctly classified
  - $\underline{a}_2^T \underline{y}_2 = [3/2\ \ 3/2][2\ \ 1]^T = 9/2 > 0$ (correct)
- Check if $\underline{y}_3$ is correctly classified
  - $\underline{a}_2^T \underline{y}_3 = [3/2\ \ 3/2][-5/2\ \ 5/2]^T = 0$ (misclassified)
- Since $\underline{y}_3$ is misclassified, we compute a_3 as

$$\underline{a}_3 = \underline{a}_2 + \underline{y}_3 = (-1, 4) \tag{41}$$

Perceptron approach: Example

- Check if $\underline{y}_1$ is correctly classified
  - $\underline{a}_3^T \underline{y}_1 = [-1 \;\; 4][4 \;\; -1]^T = 0$ (misclassified)
- Since $\underline{y}_1$ is misclassified, we compute a_4 as

$$\underline{a}_4 = \underline{a}_3 + \underline{y}_1 = (3, 3) \tag{42}$$

- Repeat until solution is reached!
- This sequential form can be viewed as reinforcement learning for machine learning.
- By combining perceptron classifiers until multi-layered networks, what we end up with are what we commonly refer to as neural networks!

Minimum squared error approach

- One issue with the perceptron approach is that if the classes are not linearly separable, the learning procedure will never stop since there will always be misclassified samples!
- One way around this is to terminate after a fixed number of iterations, but the resulting weight vector may or may not be appropriate for classification.
- Solution: What if we use a different criterion that will converge even if there are misclassified samples?
- The minimum squared error criterion provides a good compromise in performance for both separable and non-separable problems.

Minimum squared error approach

- Instead of solving a set of inequalities:

$$\underline{a}^T \underline{y}_i > 0,\ i = 1, \ldots, N \tag{43}$$

- we can obtain a solution vector for a set of equations:

$$\underline{a}^T \underline{y}_i = b_i,\ i = 1, \ldots, N \tag{44}$$

- Let the error vector $\underline{e}$ be defined as:

$$\underline{e} = \begin{bmatrix} \underline{y}_1^T \\ \vdots \\ \underline{y}_i^T \\ \vdots \\ \underline{y}_N^T \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{bmatrix} - \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_N \end{bmatrix} = Y\underline{a} - \underline{b} \tag{45}$$

Minimum squared error approach

- Instead of finding a solution $\underline{a}$ that gives no misclassifications, which could be impossible if it is not a linearly separable problem, we want to find a solution $\underline{a}$ that minimizes $|\underline{e}|^2$.
- This gives us the following sum of squared error criterion function:

$$\nabla J_s(\underline{a}) = |\underline{e}|^2 = |Y\underline{a} - \underline{b}|^2 = \sum_{i=1}^{N} (\underline{a}^T \underline{y}_i - b_i)^2 \tag{46}$$

Minimum squared error approach

- The gradient of $J_s(\underline{a})$ can be written as:

$$\nabla J_s(\underline{a}) = Y^T(Y\underline{a}_k - \underline{b}) \quad (47)$$

- This gives us the weight update formula as:

$$\underline{a}_{k+1} = \underline{a}_k + \rho_k \nabla J_p(\underline{a}) \quad (48)$$

$$\underline{a}_{k+1} = \underline{a}_k - \rho_k Y^T(Y\underline{a}_k - \underline{b}) \quad (49)$$

Minimum squared error approach

- **Step 1:** Set an initial guess for the weight vector ($\underline{a}_0$) and let $k = 0$
- **Step 2:** Determine $\underline{a}_{k+1}$ as

$$\underline{a}_{k+1} = \underline{a}_k - \rho_k Y^T (Y\underline{a}_k - \underline{b}) \tag{50}$$

- **Step 3:** If convergence reached, stop. Otherwise, go to Step 2.

Sequential variant of minimum squared error approach

- Sequential variant of minimum squared error approach:
- **Step 1:** Set an initial guess for the weight vector ($\underline{a}_0$) and let $k = 0$
- **Step 2:** Determine $\underline{a}_{k+1}$ as

$$\underline{a}_{k+1} = \underline{a}_k - \rho_k (b_k - \underline{a}_k^T \underline{y}^k) \underline{y}^k \tag{51}$$

- **Step 3:** If convergence reached, stop. Otherwise, go to Step 2.

Minimum squared error approach: parameter setting

- How do we set up parameters (i.e., ρ_k, b) for the MSE approach?
- Typically, ρ_k decreases with k (e.g., ρ_k/k) to obtain convergence
- In terms of b, useful settings include:
  - Setting $\underline{b}$ as a vector of ones
  - Setting the first N_1 of the N components to N/N_1 and the rest to N/N_2, where N_1 and N_2 are the number of samples in each class (e.g., if there are 10 samples in class 1 and 3 samples in class 2, then the first 10 components of $\underline{b}$ are set to 13/10 and the rest are set to 13/3.